

---

# Matroid Bandits: Fast Combinatorial Optimization with Learning

---

Branislav Kveton, Zheng Wen, Azin Ashkan, Hoda Eydgahi, and Brian Eriksson

Technicolor Labs

Palo Alto, CA

{branislav.kveton,zheng.wen,azin.ashkan,hoda.eydgahi,brian.eriksson}@technicolor.com

## Abstract

A matroid is a notion of independence in combinatorial optimization that characterizes problems that can be solved efficiently. In particular, it is well known that the maximum of a constrained modular function can be found greedily if and only if the constraints define a matroid. In this work, we bring together the concepts of matroids and bandits, and propose the first learning algorithm for maximizing a stochastic modular function on a matroid. The function is initially unknown and we learn it by interacting repeatedly with the environment. Our solution has two important properties. First, it is computationally efficient. In particular, its per-step time complexity is  $O(L \log L)$ , where  $L$  is the number of items in the ground set of a matroid. Second, it is provably sample efficient. Specifically, we show that the regret of the algorithm is at most linear in all constants of interest and sublinear in time. We also prove a lower bound and argue that our gap-dependent upper bound is tight. Our method is evaluated on three real-world problems and we demonstrate that it is practical.

are matroids, such as sets of linearly independent vectors and transversals of a graph.

In this paper, we propose an algorithm for learning how to maximize a stochastic modular function on a matroid. We view this problem as finding a maximum-weight basis of a matroid in expectation, where each item  $e$  in the ground set  $E$  of the matroid is associated with a stochastic weight  $w(e)$ . The weights of all items are represented as a vector  $w$ , which is drawn i.i.d. from some probability distribution  $P$ . The distribution  $P$  is initially unknown and we learn it by interacting repeatedly with the environment.

Many real-world optimization problems can be cast in our setting, such as building a spanning tree for network routing [18]. If the distribution of link delays is known, then this problem is equivalent to finding a minimum spanning tree. If the distribution is unknown, then it must be learned, perhaps while solving the problem. We examine this problem further in our experiments.

We make three contributions. First, we bring together the concepts of a matroid [23] and stochastic bandits [15, 3], and propose a learning problem of *matroid bandits*. On one hand, matroid bandits are a novel learning framework for matroids, a broad and important class of combinatorial optimization problems. On the other hand, matroid bandits are a class of  $K$ -step bandit problems that can be solved almost as efficiently as the multi-armed bandit problem, only with  $K$  times larger regret.

Second, we propose a simple greedy algorithm for solving our problem, which explores based on the optimism in the face of uncertainty. We refer to it as *Optimistic Matroid Maximization* (OMM). The algorithm has two key properties. First, it is guaranteed to be computationally efficient, because its per-step time complexity is  $O(L \log L)$ , where  $L$  is the number of items. Second, it is guaranteed to be sample efficient. More specifically, we show that the regret of the algorithm is at most linear in all constants of interest and sublinear in time. To the best of our knowledge, this is the first algorithm for a broad class of combinatorial bandits that comes with such guarantees.

## 1 Introduction

Combinatorial optimization has many applications, ranging from resource allocation [14] to routing on graphs [18]. The scale of modern problems prevents brute-force solutions, and even low-order polynomial algorithms are often impractical. Fortunately, many combinatorial optimization problems, such as finding a minimum spanning tree, can be solved greedily. These problems can be often viewed as a *matroid* [23], a generalized notion of linear independence that allows computationally efficient solutions. In particular, it is known that the maximum of a constrained modular function can be found greedily if and only if the constraints define a matroid [8]. Many mathematical objects

Finally, we evaluate our method on three real-world matroid problems. In the first problem, we learn a minimum spanning tree from Internet delay measurements. In the second problem, we learn a maximum-weight matching on a graph for microfinance funding. In the third problem, we learn a maximum-weight linearly independent set for preference elicitation. All of these problems can be solved efficiently in our framework. Therefore, we clearly demonstrate that matroid bandits are feasible in practice.

## 2 Matroids

A *matroid* is a pair  $M = (E, \mathcal{I})$ , where  $E = \{1, \dots, L\}$  is a set of  $L$  items, called the *ground set*, and  $\mathcal{I}$  is a family of subsets of  $E$ , called the *independent sets*. The family  $\mathcal{I}$  has three properties. First,  $\emptyset$  is an independent set. Second, all subsets of an independent set are independent. Finally, for all  $X \in \mathcal{I}$  and  $Y \in \mathcal{I}$  such that  $|X| = |Y| + 1$  there exists an item  $e \in X \setminus Y$  such that  $Y \cup \{e\} \in \mathcal{I}$ . This is known as the *augmentation property*. In this paper, we denote by:

$$E(X) = \{e : e \notin X, X \cup \{e\} \in \mathcal{I}\} \quad (1)$$

a set of items  $e \in E$  that can augment  $X$  such that the set remains independent.

A set  $X \in \mathcal{I}$  is a *basis* of a matroid if the set is a *maximal independent set*,  $X \cup \{e\} \notin \mathcal{I}$  for all  $e \in E \setminus X$ . All bases of a matroid are of the same size,  $K = \text{rank}(M)$ , the *rank* of the matroid [23].

A *weighted matroid* is a matroid  $M$  together with a weight function  $\mathbf{w} : E \mapsto \mathbb{R}^+$ , a function that assigns each item  $e$  in the ground set  $E$  a non-negative real number. The total weight of all items in a set  $A \subseteq E$  is defined as:

$$f(A, \mathbf{w}) = \sum_{e \in A} \mathbf{w}(e), \quad (2)$$

a modular function in  $A$ . A classical problem in combinatorial optimization is to find a maximum-weight basis of a matroid<sup>1</sup>:

$$A^* = \arg \max_{A \in \mathcal{I}} f(A, \mathbf{w}) = \arg \max_{A \in \mathcal{I}} \sum_{e \in A} \mathbf{w}(e). \quad (3)$$

It is well-known that the optimal solution  $A^*$  can be found greedily (Algorithm 1). The greedy method is also known to be optimal for all possible weight vectors  $\mathbf{w}$  if and only if  $M = (E, \mathcal{I})$  is a matroid [19]. In other words, matroids define combinatorial optimization problems that can be solved very efficiently.

## 3 Matroid Bandits

A minimum spanning tree is a maximum-weight basis of a matroid. The ground set  $E$  of this matroid are the edges of

---

**Algorithm 1** A greedy algorithm for finding a maximum-weight basis of a matroid [19].

---

**Input:** Matroid  $M = (E, \mathcal{I})$ , weights  $\mathbf{w}$

Let  $e_1, e_2, \dots, e_L$  be an ordering of items such that:

$$\mathbf{w}(e_1) \geq \mathbf{w}(e_2) \geq \dots \geq \mathbf{w}(e_L)$$

$A^* \leftarrow \emptyset$

**for all**  $i = 1, 2, \dots, L$  **do**

**if**  $(e_i \in E(A^*))$  **then**

$A^* \leftarrow A^* \cup \{e_i\}$

**end if**

**end for**

---

a graph. A set of edges is considered independent if it does not contain a cycle. Each item in the matroid is associated with a weight  $\mathbf{w}(e) = u_{\max} - u_e$ , where  $u_e$  is the weight of edge  $e$  in the graph and  $u_{\max} = \max_e u_e$ .

If the weights  $\mathbf{w}(e)$  are known, a minimum spanning tree can be computed greedily by Algorithm 1. In practice, this is not always the case. For instance, consider the problem of building a spanning tree for network routing where the delays along the links are initially unknown. In this paper, we study a variant of finding a maximum-weight basis of a matroid that can solve such problems.

### 3.1 Model

We formalize our learning problem as a matroid bandit. A *matroid bandit* is a pair  $(M, P)$ , where  $M$  is a matroid and  $P$  is a probability distribution over the weights  $\mathbf{w} \in \mathbb{R}^L$  of the items in the ground set  $E$  of  $M$ . The  $e$ -th entry of  $\mathbf{w}$ ,  $\mathbf{w}(e)$ , is the weight of item  $e$ . We assume that the weights  $\mathbf{w}$  are drawn i.i.d. from  $P$ . The mean weight is denoted by  $\bar{\mathbf{w}} = \mathbb{E}[\mathbf{w}]$  and we assume that  $\bar{\mathbf{w}}(e) \geq 0$  for all  $e \in E$ .

Each item  $e$  is associated with an *arm* and we assume that *multiple arms* can be pulled. A subset of arms  $A \subseteq E$  can be pulled if and only if  $A$  is an independent set. The return for pulling arms  $A$  is  $f(A, \mathbf{w})$  (Equation 2), the sum of the weights of all items in  $A$ . After the arms  $A$  are pulled, we observe the individual return of each arm,  $\{\mathbf{w}(e) : e \in A\}$ . This feedback model is known as *semi-bandit* [2].

We assume that the matroid  $M$  is known and the distribution  $P$  is unknown. Without loss of generality, we assume that the support of  $P$  is bounded and is a subset of  $[0, 1]^L$ . We would like to stress that we do not make any structural assumptions on  $P$ .

The optimal solution to our problem is a maximum-weight basis  $A^*$  in expectation:

$$A^* = \arg \max_{A \in \mathcal{I}} \mathbb{E}_{\mathbf{w}}[f(A, \mathbf{w})] = \arg \max_{A \in \mathcal{I}} \sum_{e \in A} \bar{\mathbf{w}}(e). \quad (4)$$

This objective is equivalent to the one in Equation 3. As a result, the optimal basis  $A^*$  in expectation can be found by a greedy algorithm (Algorithm 1).

<sup>1</sup>Without loss of generality, we assume that the optimal solution  $A^*$  is unique. It is straightforward to generalize our paper to the setting where multiple solutions are optimal.

---

**Algorithm 2** OMM: Optimistic matroid maximization.

---

**Input:** Matroid  $M = (E, \mathcal{I})$

// Initialization

Observe  $\mathbf{w}_0 \sim P$

$\hat{w}_{e,1} \leftarrow \mathbf{w}_0(e)$

$\forall e \in E$

$T_e(0) \leftarrow 1$

$\forall e \in E$

**for all**  $t = 1, 2, \dots, n$  **do**

  // Compute UCBs

$U_t(e) \leftarrow \hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)}$

$\forall e \in E$

  // Find a maximum-weight basis with respect to  $U_t$

  Let  $e_1^t, e_2^t, \dots, e_L^t$  be an ordering of items such that:

$U_t(e_1^t) \geq U_t(e_2^t) \geq \dots \geq U_t(e_L^t)$

$A^t \leftarrow \emptyset$

**for all**  $i = 1, 2, \dots, L$  **do**

**if** ( $e_i^t \in E(A^t)$ ) **then**

$A^t \leftarrow A^t \cup \{e_i^t\}$

**end if**

**end for**

  Observe  $\{\mathbf{w}_t(e) : e \in A^t\}$ , where  $\mathbf{w}_t \sim P$

  // Update statistics

$T_e(t) \leftarrow T_e(t-1)$

$\forall e \in E$

$T_e(t) \leftarrow T_e(t) + 1$

$\forall e \in A^t$

$\hat{w}_{e,T_e(t)} \leftarrow \frac{T_e(t-1)\hat{w}_{e,T_e(t-1)} + \mathbf{w}_t(e)}{T_e(t)}$

$\forall e \in A^t$

**end for**

---

Our learning problem is *episodic*. In episode  $t$ , we choose a basis  $A^t$ , according to some possibly episode-dependent policy, and gain  $f(A^t, \mathbf{w}_t)$ , where  $\mathbf{w}_t$  is the realization of the stochastic weight in episode  $t$ . Our goal is to design a policy that maximizes the *expected cumulative return* in  $n$  episodes  $\mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n}[\sum_{t=1}^n f(A^t, \mathbf{w}_t)]$ , which is equivalent to minimizing the *expected cumulative regret* in  $n$  episodes:

$$R(n) = \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[ \sum_{t=1}^n R_t(\mathbf{w}_t) \right], \quad (5)$$

where  $R_t(\mathbf{w}_t) = f(A^*, \mathbf{w}_t) - f(A^t, \mathbf{w}_t)$ .

Note that the choice of  $A^t$  impacts both the return and observations in episode  $t$ . So we need to trade off *exploration* and *exploitation*, similarly to other bandit problems.

### 3.2 Algorithm

Our solution is designed based on the *optimism in the face of uncertainty*, a principle that is employed by many bandit algorithms [3, 13, 17]. More specifically, it is a greedy method for finding a maximum-weight basis of a matroid where the expected weight  $\bar{\mathbf{w}}(e)$  of each item  $e$  is substituted for its optimistic estimate  $U_t(e)$ . Because of this, we refer to our algorithm as *Optimistic Matroid Maximization* (OMM).

The pseudocode of our method is given in Algorithm 2. In each episode  $t$ , Algorithm 2 consists of three steps. First, we compute an *upper confidence bound* (UCB) on the expected weight of each item  $e$ :

$$U_t(e) = \hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)}, \quad (6)$$

where  $\hat{w}_{e,T_e(t-1)}$  is the maximum-likelihood estimate of the weight  $\bar{\mathbf{w}}(e)$  from the first  $t-1$  episodes and initialization,  $c_{t-1,T_e(t-1)}$  is the radius of the confidence interval around this estimate, and  $T_e(t-1)$  is the number of times that item  $e$  is chosen prior to episode  $t$ .

Second, we iterate over all items  $e$ , from the highest UCB to the lowest, and greedily add them to the independent set  $A^t$ . The item  $e_i^t$  is added to  $A^t$  if it does not make the set dependent. Since our problem is a matroid, we know that the resulting set  $A^t$  is of cardinality  $K = \text{rank}(M)$ , and is a basis of the matroid  $M$ . Finally, we update the statistics  $T_e(t)$  and  $\hat{w}_{e,T_e(t)}$ .

The radius:

$$c_{t,s} = \sqrt{\frac{2 \log(t)}{s}} \quad (7)$$

is defined such that each UCB is with high probability an upper bound on the corresponding weight. The UCBs enforce exploration of items that have not been chosen very often. As the number of past episodes  $t$  increases, all confidence intervals shrink and we start exploiting most profitable items. The  $\log(t)$  term guarantees that each item is explored infinitely often as  $t \rightarrow \infty$ , to avoid linear regret.

Algorithm OMM is greedy and therefore is very fast. More specifically, suppose that the time complexity of checking for  $e_i^t \in E(A^t)$  is  $O(g(|A^t|))$ . Then the per-episode time complexity of our algorithm is  $O(L(\log(L) + g(K)))$ . In addition, we guarantee that the method behaves optimally as our estimates of  $\bar{\mathbf{w}}(e)$  become more accurate. We prove this claim in Section 4.

## 4 Analysis

Our analysis proceeds in five steps. First, we introduce our notation and basic concepts. Second, we show how to decompose the expected regret of OMM in episode  $t$  by leveraging the structure of a matroid. This is a key contribution of this paper. Third, we prove gap-dependent and gap-free upper bounds on the regret of OMM. We also derive a lower bound and argue that our gap-dependent bound is tight. Finally, we summarize the results of our analysis.

### 4.1 Notation

Before we present our results, we introduce notation used in our analysis. The optimal basis is  $A^* = \{a_1^*, \dots, a_K^*\}$ . We assume that the items in  $A^*$  are ordered such that  $a_k^*$  is the  $k$ -th item chosen by Algorithm 1. The basis chosen by

Algorithm OMM in episode  $t$  is  $A^t = \{a_1^t, \dots, a_K^t\}$ , where  $a_k^t$  denotes the  $k$ -th greedily chosen item. The hardness of discriminating item  $a_k^*$  from item  $e$  is measured by the gap between the expected weights of the items:

$$\Delta_{e,k} = \bar{\mathbf{w}}(a_k^*) - \bar{\mathbf{w}}(e) \quad (8)$$

For a given  $k$ , we denote by:

$$\mathcal{L}_k = \{e : \Delta_{e,k} > 0, e \notin A^*\} \quad (9)$$

the set of items that have a lower expected weight than  $a_k^*$  and do not belong to  $A^*$ . Finally, we denote by  $\mathbb{1}_{e,k}(t)$  the event that item  $e$  is chosen instead of item  $a_k^*$  in episode  $t$ . The expected number of times that this event happens in  $n$  episodes is:

$$Z_{e,k}(n) = \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[ \sum_{t=1}^n \mathbb{1}_{e,k}(t) \right]. \quad (10)$$

## 4.2 Regret Decomposition

A key contribution of our work is the regret decomposition of OMM that leverages the structure of a matroid.

**Theorem 1.** *The regret of OMM in episode  $t$  is bounded as:*

$$\mathbb{E}_{\mathbf{w}_t} [R_t(\mathbf{w}_t)] \leq \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \mathbb{1}_{e,k}(t). \quad (11)$$

*Proof.* Let  $A^t$  be the basis chosen in episode  $t$ . Since  $A^t$  is a basis,  $|A^*| = |A^t| = K$ , and we can rewrite the expected regret in episode  $t$  as:

$$\begin{aligned} \mathbb{E}_{\mathbf{w}_t} [R_t(\mathbf{w}_t)] &= \mathbb{E}_{\mathbf{w}_t} [f(A^*, \mathbf{w}_t) - f(A^t, \mathbf{w}_t)] \\ &= \sum_{k=1}^K \Delta_{a_k^t, \pi(k)}, \end{aligned} \quad (12)$$

where  $\pi : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$  is an arbitrary bijection from the indices in  $A^t$  to the indices in  $A^*$ .

Second, we bound the regret in episode  $t$  by leveraging the structure of a matroid. We rely on the following lemma.

**Lemma 1.** *For any two matroid bases  $A^*$  and  $A^t$ , there exists a bijection  $\pi : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$  such that:*

$$\{a_1^t, \dots, a_{k-1}^t, a_{\pi(k)}^*\} \in \mathcal{I} \quad \forall k = 1, \dots, K; \quad (13)$$

and  $\pi(k) = i$  when  $a_k^t = a_i^*$  for some  $i$ .

*Proof.* The proof of this claim is in Appendix A. ■

By definition, Algorithm OMM selects item  $a_k^t$  at step  $k$ . So the UCB of this item  $U_t(a_k^t)$  is the highest out of all items that can be chosen at step  $k$ . Based on Lemma 1, we know that  $\{a_1^t, \dots, a_{k-1}^t, a_{\pi(k)}^*\} \in \mathcal{I}$ , item  $a_{\pi(k)}^*$  can be chosen.

Because it was not, item  $a_k^t$  is chosen instead of  $a_{\pi(k)}^*$ , and the event  $\mathbb{1}_{a_k^t, \pi(k)}(t)$  happens. Furthermore, we know that  $U_t(a_k^t) \geq U_t(a_{\pi(k)}^*)$ . We leverage this fact in the proof of Lemma 2.

Finally, we are ready to decompose the expected regret:

$$\begin{aligned} \mathbb{E}_{\mathbf{w}_t} [R_t(\mathbf{w}_t)] &= \sum_{k=1}^K \Delta_{a_k^t, \pi(k)} \mathbb{1}_{a_k^t, \pi(k)}(t) \\ &\stackrel{(a)}{\leq} \sum_{k=1}^K \Delta_{a_k^t, \pi(k)} \mathbb{1}_{a_k^t, \pi(k)}(t) \mathbb{1}_{\{\Delta_{a_k^t, \pi(k)} > 0\}} \\ &\stackrel{(b)}{=} \sum_{k=1}^K \sum_{e \in \mathcal{L}_{\pi(k)}} \Delta_{e, \pi(k)} \mathbb{1}_{e, \pi(k)}(t) \\ &\stackrel{(c)}{=} \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \mathbb{1}_{e,k}(t). \end{aligned} \quad (14)$$

In step (a), we ignore all non-negative gaps. In step (b), we sum over all items  $e$  that can be chosen instead of item  $a_k^*$  such that  $\Delta_{e,k} > 0$ . In step (c), we leverage the fact that  $\pi$  is a bijection. Therefore,  $\pi(k)$  can be substituted for  $k$ . A key aspect of our decomposition is that the dependence on  $A^t$  is only in the indicator  $\mathbb{1}_{e,k}(t)$ . ■

## 4.3 Upper Bounds

Our first result is a gap-dependent bound.

**Theorem 2** (gap-dependent bound). *The expected cumulative regret of OMM is bounded as:*

$$R(n) \leq \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \frac{8}{\Delta_{e,k}} \log n + \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \left( \frac{4}{3} \pi^2 + 1 \right).$$

*Proof.* First, we bound the expected regret in episode  $t$  using Theorem 1:

$$\begin{aligned} R(n) &= \sum_{t=1}^n \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_{t-1}} [\mathbb{E}_{\mathbf{w}_t} [R_t(\mathbf{w}_t)]] \\ &\leq \sum_{t=1}^n \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_{t-1}} \left[ \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \mathbb{1}_{e,k}(t) \right] \\ &= \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \underbrace{\mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[ \sum_{t=1}^n \mathbb{1}_{e,k}(t) \right]}_{Z_{e,k}(n)}. \end{aligned} \quad (15)$$

Second, we prove an upper bound on the expected number of times when item  $e$  is chosen instead of item  $a_k^*$ .

**Lemma 2.** *For all items  $e \in \mathcal{L}_k$  and  $k \leq K$ :*

$$Z_{e,k}(n) \leq \frac{8}{\Delta_{e,k}^2} \log n + \frac{4}{3} \pi^2 + 1. \quad (16)$$

*Proof.* The proof of the lemma is in Appendix A. ■

Based on this result:

$$R(n) \leq \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} \left[ \frac{8}{\Delta_{e,k}^2} \log n + \frac{4}{3} \pi^2 + 1 \right]. \quad (17)$$

Our regret bound is obtained by reordering the terms in the above inequality. ■

We also prove a gap-free bound.

**Theorem 3** (gap-free bound). *The expected cumulative regret of OMM is bounded as:*

$$R(n) \leq K \sqrt{Ln \left[ 8 \log n + \frac{4}{3} \pi^2 + 1 \right]}.$$

*Proof.* Based on Lemma 2, the expected cumulative regret of choosing item  $e$  instead of item  $a_k^*$  is bounded as:

$$\Delta_{e,k} Z_{e,k}(n) \leq \frac{8}{\Delta_{e,k}} \log n + \Delta_{e,k} \left( \frac{4}{3} \pi^2 + 1 \right). \quad (18)$$

We multiply the inequality by  $\Delta_{e,k} Z_{e,k}(n)$  and bound the gap  $\Delta_{e,k}$  on the right-hand side by one:

$$[\Delta_{e,k} Z_{e,k}(n)]^2 \leq Z_{e,k}(n) \left[ 8 \log n + \frac{4}{3} \pi^2 + 1 \right]. \quad (19)$$

Finally, we note that  $\|\mathbf{x}\|_1 \leq \sqrt{m} \|\mathbf{x}\|_2$  for any vector  $\mathbf{x}$  of length  $m$ . Therefore:

$$\begin{aligned} R(n) &= \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} \Delta_{e,k} Z_{e,k}(n) \\ &\leq \sqrt{\sum_{k=1}^K |\mathcal{L}_k|} \sqrt{\sum_{k=1}^K \sum_{e \in \mathcal{L}_k} [\Delta_{e,k} Z_{e,k}(n)]^2} \\ &\leq \sqrt{KL \left[ 8 \log n + \frac{4}{3} \pi^2 + 1 \right]} \sum_{k=1}^K \sum_{e \in \mathcal{L}_k} Z_{e,k}(n) \\ &\leq K \sqrt{Ln \left[ 8 \log n + \frac{4}{3} \pi^2 + 1 \right]}. \end{aligned} \quad (20)$$

The last step is due to the fact that the expected number of incorrectly chosen items in  $n$  episodes is at most  $Kn$ . As a result,  $\sum_{k=1}^K \sum_{e \in \mathcal{L}_k} Z_{e,k}(n) \leq Kn$ . ■

#### 4.4 Lower Bounds

We now motivate and derive an asymptotic gap-dependent lower bound on  $R(n)$  that has similar dependence on  $n$  and gap as our gap-dependent upper bound derived in Theorem 2. Generally speaking, this lower bound is achieved

by considering a class of matroid bandits that are equivalent to  $K$  classical Bernoulli bandits, where  $K$  is the rank of the matroid. A lower bound for this class of matroid bandits can be derived based on existing lower bounds on classical bandits [15, 4].

Specifically, this lower bound is achieved with a *partition matroid*, which is defined as follows. Let  $B_1, B_2, \dots, B_K$  be a partition of  $E = \{1, \dots, L\}$ . Then the family of independent sets  $\mathcal{I}$  is defined as:

$$\mathcal{I} = \{I \subseteq E : |I \cap B_k| \leq 1, \forall k = 1, \dots, K\}. \quad (21)$$

Obviously, the rank of this partition matroid is  $K$ . We further choose the distribution  $P$  as follows:  $P$  is independent across items and  $P_e$ , the  $e$ -th marginal distribution of  $P$ , is a Bernoulli distribution with mean  $\bar{w}(e)$ . Specifically, we set:

$$\bar{w}(e) = \begin{cases} 0.5 & \text{if } e \text{ is the first item in } B_k \\ 0.5 - \epsilon & \text{otherwise} \end{cases}, \quad (22)$$

where “the first item” is the item with the smallest index, and  $\epsilon \in (0, 0.5)$  is the gap in expected rewards. We refer to this matroid bandit as  $\text{MB}^*$ . The key observation is that  $\text{MB}^*$  is equivalent to  $K$  independent Bernoulli bandits. Thus, an asymptotic lower bound for it can be derived based on existing results for classical bandits [15, 4].

To formalize the results, we need the notion of *consistent* algorithms. Specifically, a matroid bandit algorithm  $\mu$  is consistent if for any matroid bandit  $\text{MB} = (M, P)$ , any sub-optimal basis  $A$  of  $\text{MB}$ , and any  $\alpha > 0$ ,  $\mathbb{E}[T_A(n)] = o(n^\alpha)$ , where  $T_A(n)$  is the number of times basis  $A$  is chosen by episode  $n$ . Notice that restricting attention to consistent algorithms does not incur loss of generality. This is because that if an algorithm  $\mu$  is inconsistent, then there exists a matroid bandit  $\text{MB}$  such that applying  $\mu$  to  $\text{MB}$  will incur  $\Omega(n^\alpha)$  regret for some  $\alpha > 0$ . An asymptotic lower bound for the class of consistent algorithms is presented below.

**Theorem 4.**  $\forall L, K \geq 1$  s.t.  $L \geq K$ , there exists a matroid bandit  $\text{MB}^\dagger$  with  $L$  items and rank  $K$  s.t. for any consistent algorithm  $\mu$ , if  $\mu$  is applied to  $\text{MB}^\dagger$ , the following asymptotic lower bound holds:

$$\liminf_{n \rightarrow \infty} \frac{R(n)}{\log(n)} \geq \frac{L - K}{4\Delta}, \quad (23)$$

where  $\Delta = \min_k \min_{e \in \mathcal{L}_k} \Delta_{e,k}$ .

Please refer to Appendix A for the proof of Theorem 4. However, it is worth pointing out that one choice of  $\text{MB}^\dagger$  is  $\text{MB}^*$  constructed above. Obviously,  $\Delta = \epsilon$  in  $\text{MB}^*$ .

#### 4.5 Tightness of Upper Bounds

First, we argue that the regret bound in Theorem 2 is tight. More specifically, we show that it reduces to a well-known upper bound, which has a matching lower bound, for non-trivial  $L > 1$  and  $K > 1$ .

Rank $K$	Regret	Rank $K$	Regret
1	$49.60 \pm 3.11$	6	$286.87 \pm 7.91$
2	$95.64 \pm 4.43$	7	$322.60 \pm 9.56$
3	$150.23 \pm 5.34$	8	$368.84 \pm 9.24$
4	$195.30 \pm 6.49$	9	$399.35 \pm 10.45$
5	$231.91 \pm 7.11$	10	$450.47 \pm 10.59$

Table 1: The expected cumulative regret of OMM on a synthetic example from Section 4.5.

Consider a rank-2 uniform matroid on three items:

$$\begin{aligned}
E &= \{1, 2, 3\} \\
I &= \{\emptyset, \{1\}, \{2\}, \{3\}, \{1, 2\}, \{2, 3\}, \{1, 3\}\} \\
\bar{\mathbf{w}} &= (1/2, 1/3, 1/6) \\
A^* &= \{1, 2\}.
\end{aligned} \tag{24}$$

In this problem, the upper bound in Theorem 2 depends on two gaps,  $\Delta_{3,1} = 2/6$  and  $\Delta_{3,2} = 1/6$ , and resembles closely the regret bound of Auer *et al.* [3] for multi-armed bandits. This is not by chance. Indeed, our problem can be viewed as a multi-armed bandit with three arms, where the return for pulling arm  $e$  is the return for choosing all items but  $e$ . The gaps in this problem are:

$$\begin{aligned}
1/2 + 1/3 - (1/2 + 1/6) &= 1/6 \\
1/2 + 1/3 - (1/3 + 1/6) &= 2/6,
\end{aligned} \tag{25}$$

the same as  $\Delta_{2,1}$  and  $\Delta_{3,1}$ . In this case, our upper bound reduces to that of Auer *et al.* [3] and has a matching lower bound [15].

Second, we argue that the dependence on  $K$  in Theorem 2 may be intrinsic. The basis for our claim is an experiment, where we show that the regret of OMM grows linearly with  $K$ , for specific  $L$  and  $1/\Delta$ . So a good upper bound on the regret should depend on  $K$ .

Consider a uniform matroid on  $L$  items, where the weights of the items in episode  $t$  are generated as follows. First, an integer  $j$  is chosen uniformly at random from  $\{1, \dots, m\}$ . Second, for each item  $e$ , we sample  $u \sim [0, 1]$  and then set its weight as:

$$\mathbf{w}^t(e) = \mathbb{1}\{\text{mod}(e - 1, m) + 1 = j\} \mathbb{1}\{u > \varepsilon e\}. \tag{26}$$

It is easy to see that  $\bar{\mathbf{w}}(e) = (1 - \varepsilon e)/m$ . The key aspect of our design is that  $\mathbf{w}^t(i)$  and  $\mathbf{w}^t(j)$  are highly positively correlated when  $\text{mod}(i, m) = \text{mod}(j, m)$ , and highly negatively correlated otherwise. The first  $K$  items are a maximum weight basis of a uniform matroid of rank  $K$ . When  $K \leq m$ , the weights of these items are negatively correlated. In addition, when  $m \leq \sqrt{L}$  and  $\varepsilon \rightarrow 0$ , the weight of each item in the maximum-weight basis is highly correlated with the weights of at least  $K$  items that are not in the basis. In this case, it is challenging to learn the maximum-weight basis of rank  $K$ .

We choose  $L = 100$ ,  $m = 10$ , and  $\alpha = 0.001$ ; and report the regret of OMM in Table 1, in  $10^4$  episodes on 10 uniform

matroids up to rank 10. Our results confirm the hypothesis that the regret of OMM scales linearly with  $K$ . For instance, for  $K = 10$ , the regret is 9 times larger than for  $K = 1$ .

## 4.6 Summary of Theoretical Results

We proved two upper bounds on the expected cumulative regret of OMM. These bounds can be summarized as:

$$\begin{aligned}
\text{Theorem 2} \quad & O(KL(1/\Delta) \log n) \\
\text{Theorem 3} \quad & O(K\sqrt{Ln \log n}),
\end{aligned} \tag{27}$$

where  $\Delta = \min_k \min_{e \in L_k} \Delta_{e,k}$ . Both bounds are linear in  $K$ , at most linear in  $L$ , and sublinear in  $n$ . The gap-dependent bound grows much slower in  $n$  but can be loose when  $\Delta$  is small.

In Section 4.4, we prove that the regret in matroid bandits is  $\Omega(L(1/\Delta) \log n)$ . Therefore, in general, it is impossible to eliminate  $L$  in the gap-dependent bound. In Section 4.5, we identify a non-trivial variant of our problem where our upper bound matches the result of Auer *et al.* [3], which is known to be tight. As a result, our gap-dependent bound is also tight. Finally, in Section 4.5, we construct an example where the regret of OMM grows linearly with  $K$ , for specific  $L$  and  $1/\Delta$ . This empirical result provides evidence that the factor of  $K$  in our upper bounds may be intrinsic. Based on this result, we strongly believe that there exists a  $\Omega(KL(1/\Delta) \log n)$  lower bound for certain classes of matroid bandits. We leave this for future work.

## 5 Experiments

Our method is evaluated on three learning problems: learning of a minimum spanning tree (Section 5.1), learning of a maximum-weight transversal in a graph (Section 5.2), and learning of a maximum-weight linearly independent set (Section 5.3).

Each experiment consists of a series of episodes. In each episode, Algorithm OMM chooses a basis, then observes the weights of individual items in the basis, and finally updates its estimate of the world. The performance of the algorithm is measured by its *expected per-step return* in  $n$  episodes, the expected cumulative return in  $n$  episodes (Section 3.1) divided by  $n$ . This return is compared to two baselines. The first baseline is the optimal greedy solution  $A^*$  (Section 2), where all the weights are known a priori. The second baseline is an  $\varepsilon$ -greedy policy with  $\varepsilon = 0.1$ . This setting corresponds to 10% exploration and 90% exploitation, which is a standard baseline for bandit policy evaluation.

### 5.1 Minimum Spanning Tree

In the first experiment, we learn a routing network for an Internet service provider (ISP) that has the lowest total latency in expectation. We formalize this problem as learning of a minimum spanning tree. We experiment with six

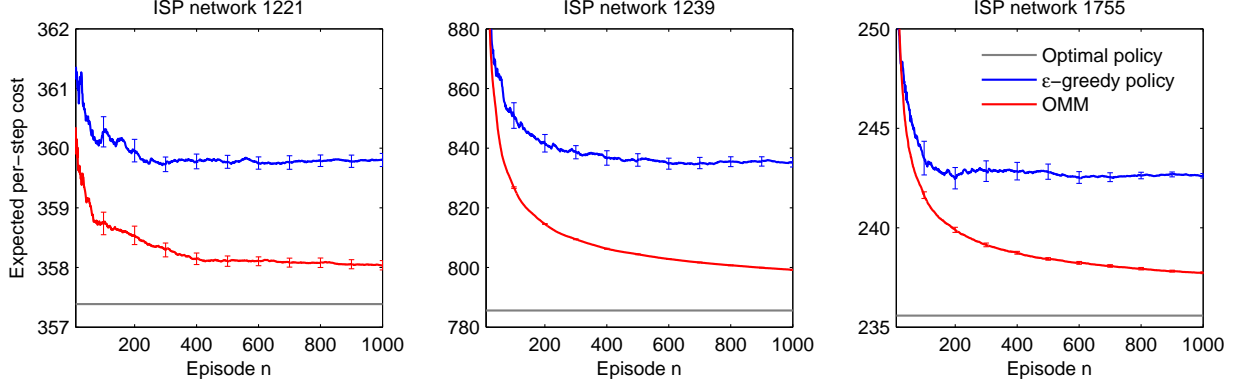


Figure 1: The expected per-step cost of building three minimum spanning trees up to episode  $n = 10^3$ .

ISP network	Number of nodes	Number of edges	Minimum latency	Maximum latency	Average latency	Optimal policy	$\epsilon$ -greedy policy	OMM
1221	108	153	1	17	2.78	357.39	$359.80 \pm 0.11$	$358.04 \pm 0.08$
1239	315	972	1	64	3.20	785.61	$835.24 \pm 1.57$	$799.21 \pm 0.11$
1755	87	161	1	31	2.91	235.58	$242.61 \pm 0.11$	$237.73 \pm 0.04$
3257	161	328	1	47	4.30	630.60	$650.03 \pm 0.52$	$638.75 \pm 0.04$
3967	79	147	1	44	5.19	345.68	$359.71 \pm 0.39$	$347.66 \pm 0.03$
6461	141	374	1	45	6.32	445.38	$495.34 \pm 1.43$	$452.27 \pm 0.07$

Table 2: The description of six ISP networks from our experiments and the expected per-step cost of building minimum spanning trees on these networks in episode  $n = 10^3$ . All latencies and costs are in milliseconds.

ISP networks from the *RocketFuel* dataset [21]. These networks have up to 300 nodes and 1k edges. The latency  $w(e)$  of edge  $e$  is modeled as a random variable with mean  $\bar{w}(e)$  and additive exponential noise  $\text{Exp}(0.5)$ . The mean latency  $\bar{w}(e)$  of each edge  $e$  is recorded in our dataset, and ranges from one to 64 milliseconds (Table 2). The reason for choosing our noise model is that most of the latency in ISP networks can be explained by geographical distances [7], the mean latency  $\bar{w}(e)$ . The noise is usually small, on the order of several hundred microseconds, and high latency due to noise is unlikely.

Our problem can be formulated as a matroid bandit. The ground set  $E$  are the edges of the network. A set of edges is independent if the edges do not form a cycle. The weight  $w(e)$  is the latency of edge  $e$ .

Our results on three ISP networks are reported in Figure 1. We observe two trends. First, the expected cost of OMM approaches that of the optimal solution  $A^*$  as the number of episodes increases. Second, OMM outperforms the  $\epsilon$ -greedy policy in just a few episodes. We report the expected costs on all networks after  $10^3$  episodes in Table 2. In all experiments, we outperform the  $\epsilon$ -greedy policy, typically by a large margin.

OMM learns relatively fast because all of our networks are sparse. In particular, the number of edges in each network is smaller than four times the number of edges in its spanning tree. So roughly speaking, each edge can be observed once in four episodes, and we can quickly learn the mean

latency of each edge.

## 5.2 Maximum Weight Matching

In the second experiment, we study the assignment of lending institutions (known as *partners*) to *lenders* in a micro-finance setting, such as Kiva [1]. Each partner is associated with a success rate which represents the probability that a loan handled by this partner is going to be paid back. The objective of the assignment is to maximize the overall success rate of the selected partners. This problem can be cast as learning a maximum weight matching in a bipartite graph, and it can be formulated under a family of matroids, called *transversal* matroids [9]. The ground set  $E$  of a transversal matroid is the set of left vertices of the corresponding bipartite graph, and the independence set  $\mathcal{I}$  are all possible matchings in the graph such that no two edges in the matching share an endpoint. In the weighted instance of the transversal matroid,  $\bar{w}(e)$  is the weight associated with the left vertices of the bipartite graph. The goal is to find a matching that maximizes the overall weight of selected vertices.

We used a sample of 194,876 loans from the Kiva micro-finance dataset [1], and created a bipartite graph. Every loan is handled by a partner (Figure 2-a). There are a total of 232 partners in the dataset that represent the left vertices of the bipartite graph. The weight  $\bar{w}(e)$  is the *mean* success rate of a partner. We estimate it from the dataset as  $\bar{w}(e) = \frac{1}{n_l} \sum_{i=1}^{n_l} w_i(e)$ , where  $n_l$  is the number of loans

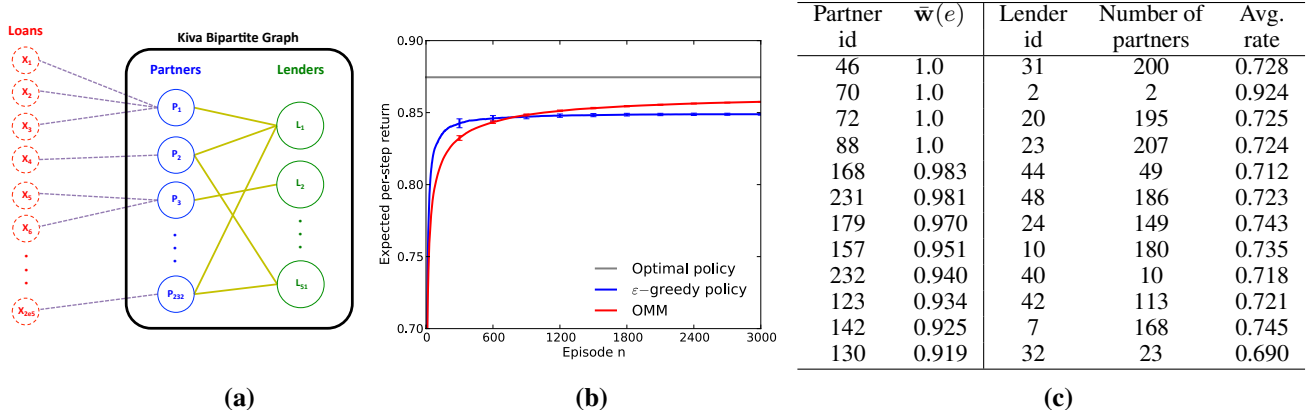


Figure 2: **(a)** The Kiva dataset can be modeled as a bipartite graph connecting lenders to field partners, which, in turn, fund several loans in the region. **(b)** The expected per-step return of finding maximum weight matching up to episode  $n = 3k$ . **(c)** Top 12 selected partners assigned based on their mean success rate in the optimal solution  $A^*$ . The optimal solution involves 46 partner/lender assignments.

handled by this partner. We assume  $w_i(e)$  is 0.7 if the loan  $i$  is in repayment, 1 if it is paid, and 0 otherwise. There are 286,874 lenders in the dataset. We grouped these lenders into 51 clusters according to their location: 50 representing each individual state in United States, and 1 representing all foreign lenders. These 51 lender clusters constitute the right vertices of the bipartite graph, and are referred to as lenders in the remainder of the paper. There is an edge between a partner and a lender if the lender is among the top 50% supporters of the partner (in terms of the number of funded loans for this partner), resulting in approximately  $5k$  edges in the bipartite graph. In each episode, the success rate of a partner,  $w(e)$ , is estimated as the success rate of one of its associated loans chosen at random.

The optimal solution  $A^*$  finds a matching in the graph that maximizes the overall success rate of the selected partners. Top twelve partners selected based on their mean success rate in the optimal solution are shown in Figure 2-c. For each partner, the id of the lender to which this partner was assigned along with the number of eligible partners of the lender and their average success rate are listed in the Table. The objective of OMM and  $\epsilon$ -greedy policies is similar to the optimal policy with the difference that success rates (i.e.  $w(e)$ ) are not known beforehand, and they must be learned by interacting repeatedly with the environment. Comparison results of the three policies are reported in Figure 2-b. Similar to the previous experiment, we observe the following trends. First, the expected return of OMM approaches that of the optimal solution  $A^*$  as the number of episodes increases. Second, OMM outperforms the  $\epsilon$ -greedy policy.

### 5.3 Maximum-Weight Linearly Independent Set

In the third experiment, we study a preference elicitation problem in a movie recommendation domain. We cast the problem as learning a set of movies, which are rated by

people to express their preferences. We want these movies to be popular, so the people know them and can rate them. We also want the movies to be diverse, so we learn about the different aspects of people’s preferences.

Our problem can be formulated as a matroid bandit. The ground set  $E$  is a set of movies. We experiment with 100 most rated movies in the *MovieLens* dataset [16], a dataset of 6k people who rated one million movies. A set of movies is independent if none of the movies in the set can be written as a linear combination of the genres of the other movies. This is our notion of diversity. The weight  $\bar{w}(e)$  is the probability that movie  $e$  is known. We estimate it as  $\bar{w}(e) = \frac{1}{n_p} \sum_{i=1}^{n_p} w_i(e)$ , where  $n_p$  is the number of people in our dataset and  $w_i(e)$  is an indicator of person  $i$  rating movie  $e$ . In each episode, we choose the person  $w_i$  randomly. Our goal is to maximize the expected number of movies that are independent and known by the person.

Twelve most popular movies from the optimal solution  $A^*$  are shown in Figure 3. These movies range from *Comedy* to *War* and *Sci-fi*, and are quite diverse. This validates our assumption that linear independence can be used to represent diversity. Our learning results are reported in the same figure. We observe two trends. First, the expected return of OMM approaches that of the optimal solution  $A^*$  as the number of episodes increases. Second, we outperform the  $\epsilon$ -greedy policy after 12k episodes.

## 6 Related Work

The problem of combinatorial bandits has been studied extensively in recent years [11, 6, 5, 2]. The existing work on this topic can be divided into two categories, the stochastic and adversarial setting.

Stochastic combinatorial bandits were pioneered by Gai *et al.* [11]. They proposed a UCB-style algorithm and proved



k	Movie name	$\bar{w}(e)$	Movie genres
1	American Beauty	0.568	Comedy, Drama
2	Jurassic Park	0.442	Action, Adventure, Sci-Fi
3	Saving Private Ryan	0.439	Action, Drama, War
4	The Matrix	0.429	Action, Sci-Fi, Thriller
5	Back to the Future	0.428	Comedy, Sci-Fi
6	The Silence of the Lambs	0.427	Drama, Thriller
7	Men in Black	0.420	Action, Adventure, Comedy, Sci-Fi
8	Fargo	0.416	Crime, Drama, Thriller
9	Shakespeare in Love	0.392	Comedy, Romance
10	L.A. Confidential	0.379	Crime, Film-Noir, Mystery, Thriller
11	E.T.	0.376	Children's Drama, Fantasy, Sci-Fi
12	Ghostbusters	0.361	Comedy, Horror

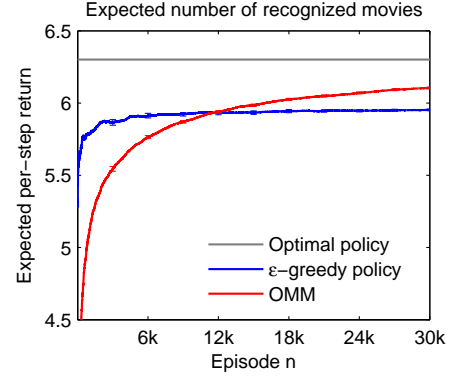


Figure 3: **Left.** Twelve most popular movies in the optimal solution  $A^*$ . The optimal solution involves 17 movies. **Right.** The expected per-step return of three preference elicitation policies up to episode  $n = 30k$ .

that its regret is  $O(K^3 L(1/\Delta^2) \log n)$ . Chen *et al.* [6] improved the analysis of Gai *et al.* [11] and showed that the regret is  $O(K^2 L(1/\Delta) \log n)$ . Our work can be viewed as a special class of a stochastic combinatorial bandit, where the feasible set are the independent sets of a matroid. Our gap-dependent regret bound is  $O(KL(1/\Delta) \log n)$ , which is a factor of  $K$  tighter than the best of the aforementioned results.

COMBAND [5] and OSDM [2] are two popular classes of methods for adversarial combinatorial bandits. The regret of OSDM in the semi-bandit setting is  $O(\sqrt{KLn})$  and it is known to match the mini-max lower bound [2]. The main limitation of both methods is that they are not guaranteed to be computationally efficient. In particular, COMBAND may need to sample from a distribution over exponentially many solutions and OSDM needs to project to the convex hull of these solutions. In comparison, our method is computationally efficient. More specifically, the time complexity of OMM in episode  $t$  is  $O(L \log L)$ . Our gap-free regret bound is  $O(K\sqrt{Ln \log n})$ , looser by a factor of  $\sqrt{K \log n}$  when compared to the regret bound of OSDM. The factor  $\sqrt{\log n}$  can be eliminated by more careful analysis.

Matroids are a broad and important class of combinatorial optimization problems [19], and they have been an active area of research for the past 80 years. This is the first paper that formulates a well-known matroid problem, finding a maximum-weight basis of a matroid, as a bandit problem and proposes an efficient algorithm for solving it.

Matroids are a form of submodularity [19]. Submodularity is a popular topic in machine learning and several algorithms for learning how to maximize submodular functions have been proposed [12, 24, 10, 22]. All of these methods take advantage of the fact that being greedy is near optimal and mimic the greedy policy for solving the corresponding problem. In this aspect, our method is similar. However, none of these approaches can solve our problem out-of-the-box and particularly not with guarantees that come with our analysis. Our analysis takes advantage of the properties of

a matroid, and is very novel in this aspect.

## 7 Conclusions

This is the first work that studies the problem of learning a maximum-weight basis of a matroid, where the weights of the items are initially unknown, and have to be learned by interacting repeatedly with the environment. We propose a practical algorithm for solving the problem and analyze its expected cumulative regret. We show that the regret grows sublinearly with time and is at most linear in all quantities of interest. Finally, we evaluate our proposed algorithm on three real-world problems and show that it is practical.

Our regret bounds scale linearly with the number of items  $L$  and the time complexity of OMM is  $\Omega(L)$ . Therefore, our approach is not practical when  $L$  is large. We believe that such problems can be solved efficiently by introducing additional structure. For instance, suppose that the expected weight of items is a function of features that are associated with each item. Then our problem reduces to learning one regressor, as opposing to  $L$  separate weights. We leave this for future work.

Many combinatorial optimization problems can be viewed as optimization on matroids [20]. In a sense, these are the hardest combinatorial problems that can be solved in polynomial time. In this work, we showed that one of these problems, maximization of a modular function on a matroid, is efficiently learnable in the bandit setting. The key idea in our analysis, regret decomposition based on the structure of a matroid, is very general, and we believe that it can be applied broadly in the design and analysis of efficient learning algorithms for other combinatorial optimization problems. One such problem is *maximum-weight matching* on a bipartite graph, which is an instance of maximizing a modular function on the intersection of two matroids. Another problem is finding a *minimum-cost max-flow*, which is an instance of maximizing a modular function on a *polymatroid*, a generalization of a matroid.

## References

- [1] KIVA. <http://build.kiva.org/docs/data>, 2013.
- [2] Jean-Yves Audibert, Sebastien Bubeck, and Gabor Lugosi. Regret in online combinatorial optimization. *Mathematics of Operations Research*, 39(1):31–45, 2014.
- [3] Peter Auer, Nicolo Cesa-Bianchi, and Paul Fischer. Finite-time analysis of the multiarmed bandit problem. *Machine Learning*, 47:235–256, 2002.
- [4] Sébastien Bubeck and Nicolò Cesa-Bianchi. Regret analysis of stochastic and nonstochastic multi-armed bandit problems. *Foundations and Trends in Machine Learning*, 5(1):1–122, 2012.
- [5] Nicolò Cesa-Bianchi and Gábor Lugosi. Combinatorial bandits. *Journal of Computer and System Sciences*, 78(5):1404–1422, 2012.
- [6] Wei Chen, Yajun Wang, and Yang Yuan. Combinatorial multi-armed bandit: General framework and applications. In *Proceedings of the 30th International Conference on Machine Learning*, pages 151–159, 2013.
- [7] Baek-Young Choi, Sue Moon, Zhi-Li Zhang, Konstantina Papagiannaki, and Christophe Diot. Analysis of point-to-point packet delay in an operational network. In *Proceedings of the 23rd Annual Joint Conference of the IEEE Computer and Communications Societies*, 2004.
- [8] Jack Edmonds. Matroids and the greedy algorithm. *Mathematical Programming*, 1(1):127–136, 1971.
- [9] Jack Edmonds and Delbert Ray Fulkerson. Transversals and matroid partition. *Journal of Research of the National Bureau of Standards*, 69B(3):147–153, 1965.
- [10] Victor Gabillon, Branislav Kveton, Zheng Wen, Brian Eriksson, and S. Muthukrishnan. Adaptive submodular maximization in bandit setting. In *Advances in Neural Information Processing Systems 26*, pages 2697–2705, 2013.
- [11] Yi Gai, Bhaskar Krishnamachari, and Rahul Jain. Combinatorial network optimization with unknown variables: Multi-armed bandits with linear rewards and individual observations. *IEEE/ACM Transactions on Networking*, 20(5):1466–1478, 2012.
- [12] Andrew Guillory and Jeff Bilmes. Online submodular set cover, ranking, and repeated active learning. In *Advances in Neural Information Processing Systems 24*, pages 1107–1115, 2011.
- [13] Thomas Jaksch, Ronald Ortner, and Peter Auer. Near-optimal regret bounds for reinforcement learning. *Journal of Machine Learning Research*, 11:1563–1600, 2010.
- [14] Naoki Katoh. Combinatorial optimization algorithms in resource allocation problems. *Encyclopedia of Optimization*, pages 259–264, 2001.
- [15] T.L. Lai and Herbert Robbins. Asymptotically efficient adaptive allocation rules. *Advances in Applied Mathematics*, 6(1):4 – 22, 1985.
- [16] Shyong Lam and Jon Herlocker. MovieLens 1M Dataset. <http://www.grouplens.org/node/12>, 2012.
- [17] Rémi Munos. The optimistic principle applied to games, optimization, and planning: Towards foundations of Monte-Carlo tree search. *Foundations and Trends in Machine Learning*, 2012.
- [18] Carlos AS Oliveira and Panos M Pardalos. A survey of combinatorial optimization problems in multicast routing. *Computers and Operations Research*, 32(8):1953–1981, 2005.
- [19] James Oxley. *Matroid Theory*. Oxford University Press, New York, NY, 2011.
- [20] Christos Papadimitriou and Kenneth Steiglitz. *Combinatorial Optimization*. Dover Publications, Mineola, NY, 1998.
- [21] Neil Spring, Ratul Mahajan, and David Wetherall. Measuring isp topologies with rocketfuel. *ACM SIGCOMM Computer Communication Review*, 32(4):133–145, 2002.
- [22] Zheng Wen, Branislav Kveton, Brian Eriksson, and Sandilya Bhamidipati. Sequential Bayesian search. In *Proceedings of the 30th International Conference on Machine Learning*, pages 977–983, 2013.
- [23] Hassler Whitney. On the abstract properties of linear dependence. *American Journal of Mathematics*, 57(3):509–533, 1935.
- [24] Yisong Yue and Carlos Guestrin. Linear submodular bandits and their application to diversified retrieval. In *Advances in Neural Information Processing Systems 24*, pages 2483–2491, 2011.

## A Technical Lemmas

### A.1 Proof for Lemma 1

For convenience, we restate the lemma below.

**Lemma 3.** *For any two matroid bases  $A^*$  and  $A^t$ , there exists a bijection  $\pi : \{1, \dots, K\} \rightarrow \{1, \dots, K\}$  such that:*

$$\left\{a_1^t, \dots, a_{k-1}^t, a_{\pi(k)}^*\right\} \in \mathcal{I} \quad \forall k = 1, \dots, K; \quad (28)$$

and  $\pi(k) = i$  when  $a_k^t = a_i^*$  for some  $i$ .

*Proof.* Our proof is constructive. The central idea is to exchange items in  $A^t$  for items in  $A^*$  in backward order, from  $a_K^t$  to  $a_1^t$ .

We start with item  $a_K^t$ . From the augmentation property, there exists  $a_i^* \in A^* - (A^t - a_K^t)$  such that  $A^t - a_K^t + a_i^* \in \mathcal{I}$ . Let one of  $a_i^*$  be such that  $a_i^* = a_{K-1}^t$ . Then we set  $\pi(K) = i$ . Otherwise, we select any  $a_i^*$  and set  $\pi(K) = i$ . The result of this step is an independent set  $B_{K-1} = \left\{a_1^t, \dots, a_{K-1}^t, a_{\pi(K)}^*\right\} \in \mathcal{I}$ .

We continue with item  $a_{K-1}^t$ . From the augmentation property, there exists item  $a_i^* \in A^* - (B_{K-1} - a_{K-1}^t)$  such that  $B_{K-1} - a_{K-1}^t + a_i^* \in \mathcal{I}$ . Let one of  $a_i^*$  be such that  $a_i^* = a_{K-2}^t$ . Then we set  $\pi(K-1) = i$ . Otherwise, we choose any  $a_i^*$  and set  $\pi(K-1) = i$ . The result of this step is an independent set  $B_{K-2} = \left\{a_1^t, \dots, a_{K-2}^t, a_{\pi(K-1)}^*, a_{\pi(K)}^*\right\} \in \mathcal{I}$ .

The same construction applies to item  $a_{K-2}^t$ , all the way down to item  $a_1^t$ . The result is a sequence of independent sets:

$$B_{k-1} = \left\{a_1^t, \dots, a_{k-1}^t, a_{\pi(k)}^*, \dots, a_{\pi(K)}^*\right\} \in \mathcal{I} \quad \forall k = 1, \dots, K \quad (29)$$

such that  $\pi(k) = i$  when  $a_k^t = a_i^*$  for some  $i$ . The claim of the lemma follows directly from the hereditary property. ■

### A.2 Proof for Lemma 2

Lemma 2 is restated below.

**Lemma 4.** *For all items  $e \in \mathcal{L}_k$  and  $k \leq K$ :*

$$\mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[ \sum_{t=1}^n \mathbb{1}_{e,k}(t) \right] \leq \frac{8}{\Delta_{e,k}^2} \log n + \frac{4}{3} \pi^2 + 1. \quad (30)$$

*Proof.* Our proof has the same structure as the proof of Theorem 1 by Auer *et al.* [3]. Let  $\ell_{e,k}$  be a positive integer. Then for all items  $e \in \mathcal{L}_k$  and  $k$ :

$$\begin{aligned} \sum_{t=1}^n \mathbb{1}_{e,k}(t) &\leq \ell_{e,k} + \sum_{t=\ell_{e,k}+1}^n \mathbb{1}_{e,k}(t) \mathbb{1}\{T_e(t-1) > \ell_{e,k}\} \\ &\leq \ell_{e,k} + \sum_{t=\ell_{e,k}+1}^n \mathbb{1}\{\hat{w}_{e,T_e(t-1)} + c_{t-1,T_e(t-1)} \geq \\ &\quad \hat{w}_{a_k^*, T_{a_k^*}(t-1)} + c_{t-1, T_{a_k^*}(t-1)}, T_e(t-1) > \ell_{e,k}\} \\ &\leq \ell_{e,k} + \sum_{t=\ell_{e,k}+1}^n \sum_{s=1}^t \sum_{s_e=\ell_{e,k}+1}^t \mathbb{1}\{\hat{w}_{e,s_e} + c_{t-1,s_e} \geq \hat{w}_{a_k^*,s} + c_{t-1,s}\} \\ &= \ell_{e,k} + \sum_{t=\ell_{e,k}}^{n-1} \sum_{s=1}^{t+1} \sum_{s_e=\ell_{e,k}+1}^{t+1} \mathbb{1}\{\hat{w}_{e,s_e} + c_{t,s_e} \geq \hat{w}_{a_k^*,s} + c_{t,s}\} \\ &\leq \ell_{e,k} + \sum_{t=1}^{\infty} \sum_{s=1}^{t+1} \sum_{s_e=\ell_{e,k}}^{t+1} \mathbb{1}\{\hat{w}_{e,s_e} + c_{t,s_e} \geq \hat{w}_{a_k^*,s} + c_{t,s}\}. \end{aligned} \quad (31)$$

When  $\hat{w}_{e,s_e} + c_{t,s_e} \geq \hat{w}_{a_k^*,s} + c_{t,s}$ , at least one of the following events must happen:

$$\hat{w}_{a_k^*,s} \leq \bar{\mathbf{w}}(a_k^*) - c_{t,s} \quad (32)$$

$$\hat{w}_{e,s_e} \geq \bar{\mathbf{w}}(e) + c_{t,s_e} \quad (33)$$

$$\bar{\mathbf{w}}(a_k^*) < \bar{\mathbf{w}}(e) + 2c_{t,s_e}. \quad (34)$$

We bound the probability of the first two events (Equations 32 and 33) using Hoeffding's inequality:

$$P(\hat{w}_{a_k^*,s} \leq \bar{\mathbf{w}}(a_k^*) - c_{t,s}) \leq \exp[-4 \log t] = t^{-4} \quad (35)$$

$$P(\hat{w}_{e,s_e} \geq \bar{\mathbf{w}}(e) + c_{t,s_e}) \leq \exp[-4 \log t] = t^{-4}. \quad (36)$$

When  $\ell_{e,k} = \left\lceil \frac{8}{\Delta_{e,k}^2} \log n \right\rceil$ , the third event (Equation 34) cannot happen. More specifically, for all  $s_e \geq \frac{8}{\Delta_{e,k}^2} \log n$ :

$$\bar{\mathbf{w}}(a_k^*) - \bar{\mathbf{w}}(e) - 2c_{t,s_e} = \Delta_{e,k} - 2\sqrt{\frac{2 \log t}{s_e}} \geq 0. \quad (37)$$

Therefore, we may conclude that:

$$\begin{aligned} \mathbb{E}_{\mathbf{w}_1, \dots, \mathbf{w}_n} \left[ \sum_{t=1}^n \mathbb{1}_{e,k}(t) \right] &\leq \ell_{e,k} + \sum_{t=1}^{\infty} \sum_{s=1}^{t+1} \sum_{s_e=\ell_{e,k}}^{t+1} [P(\hat{w}_{a_k^*,s} \leq \bar{\mathbf{w}}(a_k^*) - c_{t,s}) + P(\hat{w}_{e,s_e} \geq \bar{\mathbf{w}}(e) + c_{t,s_e})] \\ &\leq \ell_{e,k} + \sum_{t=1}^{\infty} 2(t+1)^2 t^{-4} \\ &\leq \ell_{e,k} + \sum_{t=1}^{\infty} 8t^{-2} \\ &= \ell_{e,k} + \frac{4}{3} \pi^2. \end{aligned} \quad (38)$$

■

### A.3 Proof for Theorem 4

The proof for Theorem 4 is presented below.

*Proof.* For any  $l, k \geq 1$  s.t.  $L \geq K$ , we construct a partition matroid  $M^*$  as follows. Let  $B_1, B_2, \dots, B_K$  be an arbitrary but fixed partition of  $E = \{1, 2, \dots, L\}$ , then  $\mathcal{I}$ , the family of independent sets, is defined as

$$\mathcal{I} = \{I \subseteq E : |I \cap B_k| \leq 1, \forall k = 1, \dots, K\},$$

and we choose  $M^* = (E, \mathcal{I})$ . We also define  $\mathcal{P}$ , a class of probability distributions over  $\{0, 1\}^L$ , as

$$\mathcal{P} = \{P : P \text{ is a distribution over } \{0, 1\}^L \text{ and independent across items}\}.$$

Notice that  $\forall P \in \mathcal{P}$ , the marginal distribution  $P_e$ 's are independent Bernoulli distributions. We define a class of Bernoulli partition matroid bandits as  $\mathcal{M} = \{(M^*, P) : P \in \mathcal{P}\}$ .

For any consistent algorithm  $\mu$ , and any matroid bandit  $\text{MB} \in \mathcal{M}$ , if  $\mu$  is applied to  $\text{MB}$ , then by definition, for any sub-optimal basis  $A$  of  $\text{MB}$  and any  $\alpha > 0$ , we have  $\mathbb{E}[T_A(n)] = o(n^\alpha)$ . This further implies that for any  $e \in E$  that does not belong to any optimal basis, we must have  $\mathbb{E}[T_e(n)] = o(n^\alpha)$ . To see it, notice that  $T_e(n) = \sum_{A: e \in A} T_A(n)$ , thus  $\forall \alpha > 0$ , we have

$$\lim_{n \rightarrow \infty} \frac{\mathbb{E}[T_e(n)]}{n^\alpha} = \sum_{A: e \in A} \lim_{n \rightarrow \infty} \frac{\mathbb{E}[T_A(n)]}{n^\alpha} = 0,$$

where the second equality follows from the facts that  $A$  is sub-optimal  $\forall A \ni e$  and algorithm  $\mu$  is consistent.

By definition of  $\mathcal{M}$ , any  $\text{MB} \in \mathcal{M}$  is equivalent to  $K$  Bernoulli bandits with  $|B_1|, |B_2|, \dots, |B_K|$  arms. On the other hand, any  $K$  Bernoulli bandits with  $|B_1|, |B_2|, \dots, |B_K|$  arms can be represented by a matroid bandit  $\text{MB} \in \mathcal{M}$ . Hence the above

result implies that for any consistent algorithm  $\mu$ , any  $K$  Bernoulli bandits with  $|B_1|, |B_2|, \dots, |B_K|$  arms, and any set of Bernoulli reward distributions, we have  $\mathbb{E}[T_e(n)] = o(n^\alpha)$  for any sub-optimal arm<sup>2</sup>  $e$  and any  $\alpha > 0$ . From Theorem 2.2 of [4], if we apply a consistent algorithm  $\mu$  to  $\text{MB}^*$  defined in Section 4.4, then we have

$$\begin{aligned}
\liminf_{n \rightarrow \infty} \frac{R(n)}{\log(n)} &\stackrel{(a)}{\geq} \sum_{k=1}^K \sum_{e \in B_k: \bar{\mathbf{w}}(e) < 0.5} \frac{\epsilon}{\text{kl}(0.5 - \epsilon, 0.5)} \\
&= \sum_{k=1}^K \frac{(|B_k| - 1) \epsilon}{\text{kl}(0.5 - \epsilon, 0.5)} = \frac{(L - K) \epsilon}{\text{kl}(0.5 - \epsilon, 0.5)} \\
&\stackrel{(b)}{\geq} \frac{(L - K)}{4\epsilon} \stackrel{(c)}{=} \frac{(L - K)}{4\Delta},
\end{aligned} \tag{39}$$

where  $\text{kl}(0.5 - \epsilon, 0.5)$  is the Kullback-Leibler divergence between Bernoulli parameters  $0.5 - \epsilon$  and  $0.5$ . Specifically, inequality (a) follows from Theorem 2.2 of [4], inequality (b) follows from the inequality  $\text{kl}(p, q) \leq \frac{(p-q)^2}{q(1-q)}$  for any  $p, q \in (0, 1)$ , and equality (c) follows from the fact that  $\Delta = \epsilon$  in  $\text{MB}^*$ . ■

---

<sup>2</sup>Assume  $e \in B_k$ , then arm  $e$  is sub-optimal if and only if  $\bar{\mathbf{w}}(e) < \max_{e' \in B_k} \bar{\mathbf{w}}(e')$ .